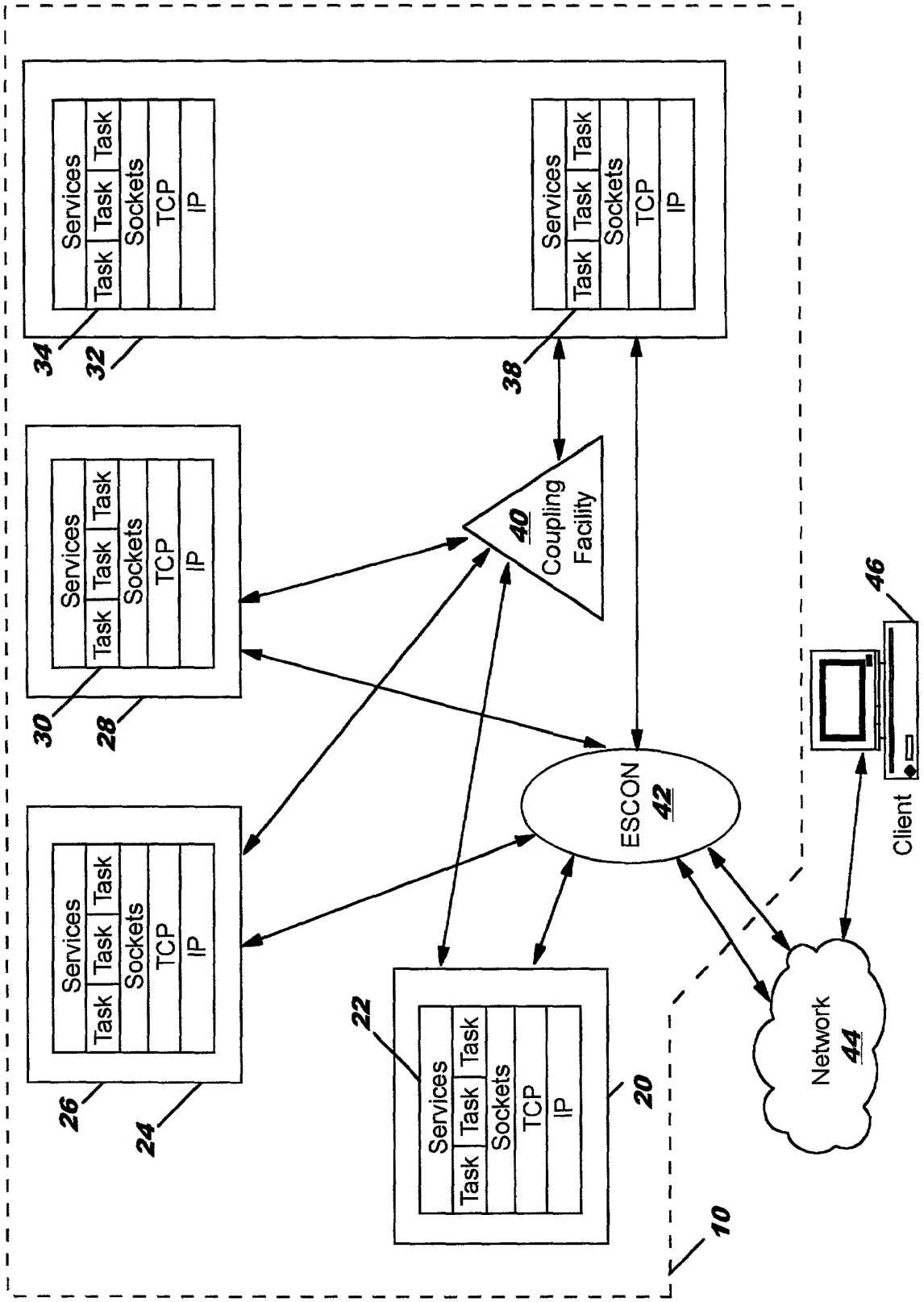
FIG. 1


Services
Task
Task
Task
Sockets
TCP
IP
34
32
38
30
28
26
24
22
20
10
40
Coupling Facility
ESCON
42
Network
44
46
Client

TOEHO T20S2860



200

| local IP address 202 | local port number 204 | partner IP address 206 | partner mask/ prefix 208 | partner port number 210 | duration 212 | socket 214 | return code 216 | addl. info. 218 |
|---|---|---|---|---|---|---|---|---|

FIG. 2A

220

| local IP address 222 | local port number 224 | partner IP address 226 | partner mask/ prefix 228 | partner port number 230 | stack identity 232 | return code 234 | addl. info. 236 |
|---|---|---|---|---|---|---|---|

FIG. 2B

TOEOHO" T2052860

240

| local IP address 242 | local port number 244 | partner IP address 246 | partner mask/ prefix 248 | partner port number 250 | duration 252 | socket 254 | return code 256 | addl. info. 258 |
|---|---|---|---|---|---|---|---|---|

FIG. 2C

260

| local IP address 262 | local port number 264 | partner IP address 266 | partner mask/ prefix 268 | partner port number 270 | stack identity 272 | return code 274 | addl. info. 276 |
|---|---|---|---|---|---|---|---|

FIG. 2D

TOEHO T20928A0

280

```
IPCONFIG SYSPLEX ROUTING DATAGRAMFWD DYNAMICXCF 193.9.200.4 255.255.255.240 1
VIPADYNAMIC
    VIPADEFINE 255.255.255.192 9.67.240.02
    VIPADISTRIBUTE  DEFINE 9.67.240.02  PORT 20 21 8000 9000  DESTIP   AUTOAFFINITY
            193.9.200.2
            193.9.200.4
            193.9.200.6
ENDVIPADYNAMIC
```

282

FIG. 2E

290

```
PORT
    20   TCP OMVS  NOAUTOLOG  AUTOAFFINITY ; FTP server data port
```

292

FIG. 2F

300

| server application type 305 | | | client affinity 325 | | |
|---|---|---|---|---|---|
| destination IP address 310 | destination port number 315 | receiving/ owning stack 320 | source IP address 330 | mask/ prefix 335 | source port number(s) 340 |

FIG. 3A

350

| server application type 355 | | | client affinity 375 | | | |
|---|---|---|---|---|---|---|
| destination IP address 360 | destination port number 365 | receiving/ owning 370 application | source IP address 380 | mask/ prefix 385 | source port 390 number(s) | timeout information 395 |

FIG. 3B

FIG. 4

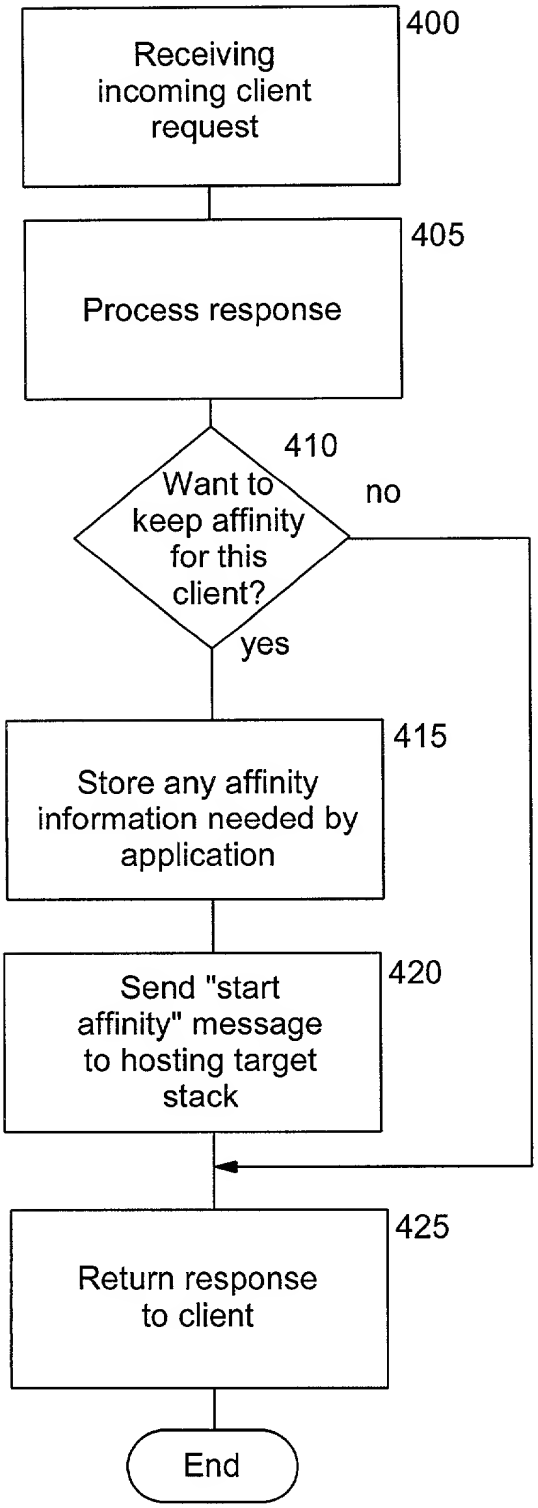

Receiving incoming client request
400
Process response
405
Want to keep affinity for this client?
410
no
yes
Store any affinity information needed by application
415
Send "start affinity" message to hosting target stack
420
Return response to client
425
End

FIG. 5

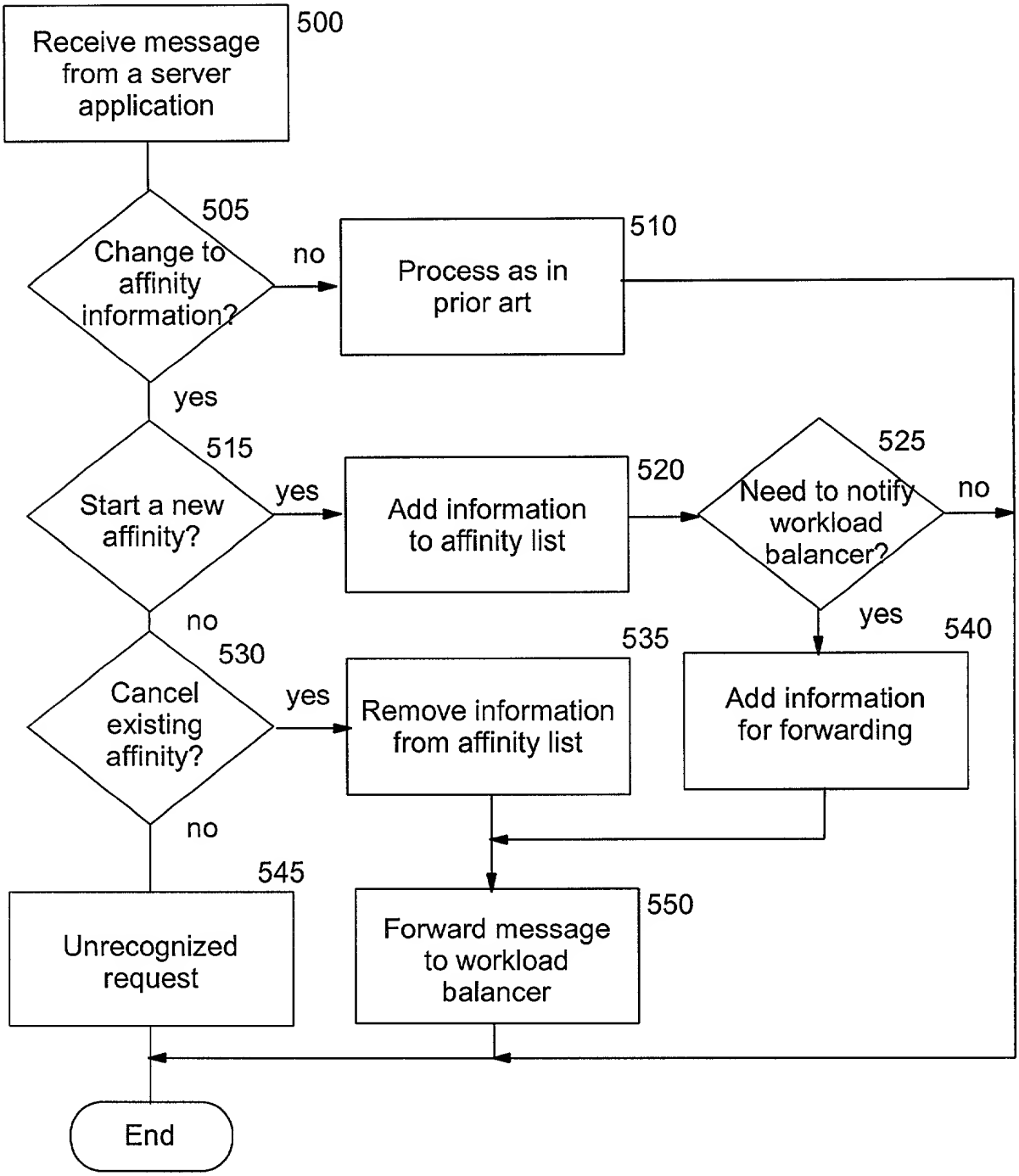

Receive message from a server application
500
Change to affinity information?
505
no
Process as in prior art
510
yes
Start a new affinity?
515
yes
Add information to affinity list
520
Need to notify workload balancer?
525
no
yes
Add information for forwarding
540
no
Cancel existing affinity?
530
yes
Remove information from affinity list
535
no
Unrecognized request
545
Forward message to workload balancer
550
End

FIG. 6

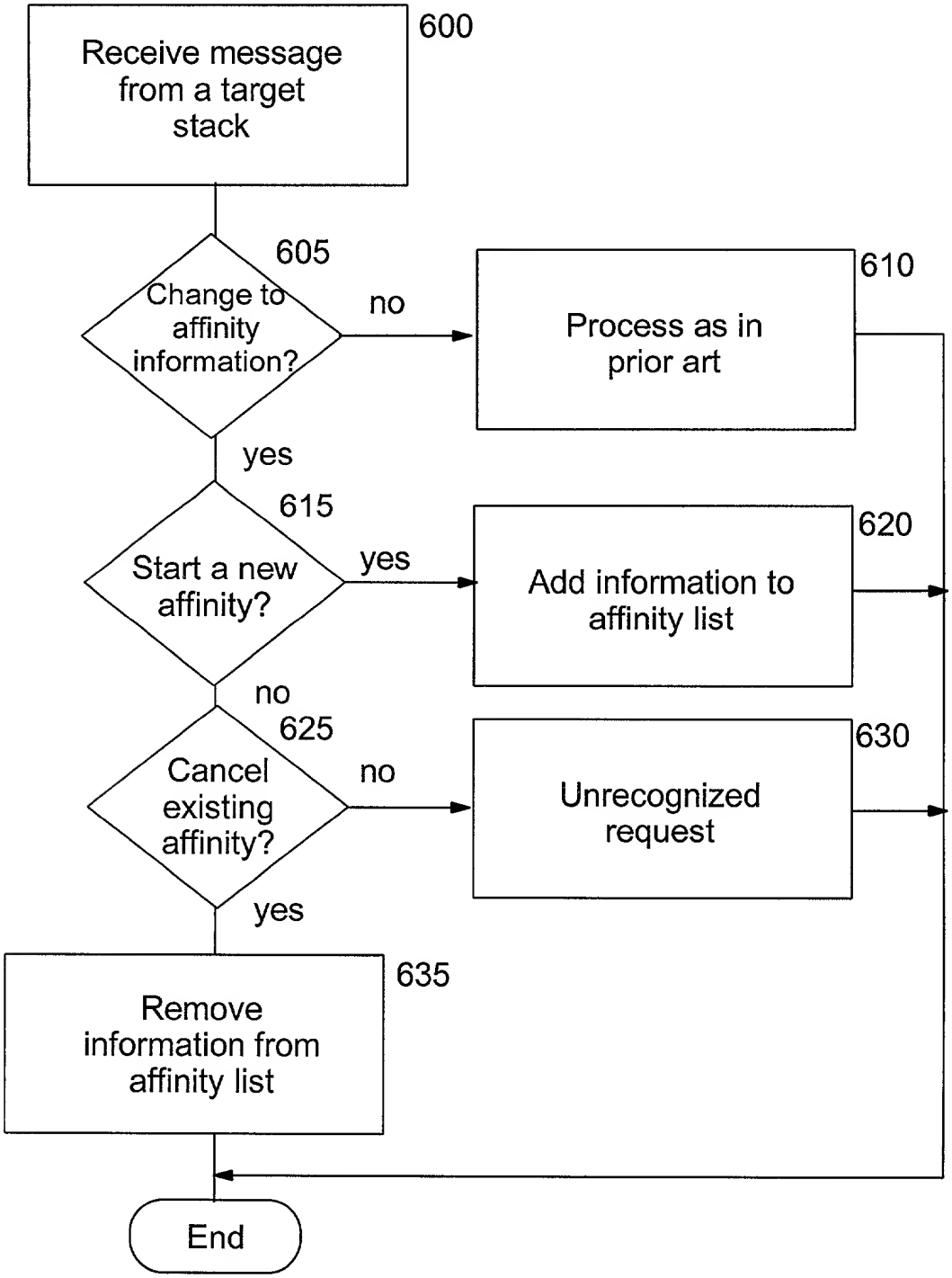

Receive message from a target stack
600
Change to affinity information?
605
no
Process as in prior art
610
yes
Start a new affinity?
615
yes
Add information to affinity list
620
no
Cancel existing affinity?
625
no
Unrecognized request
630
yes
Remove information from affinity list
635
End

FIG. 7

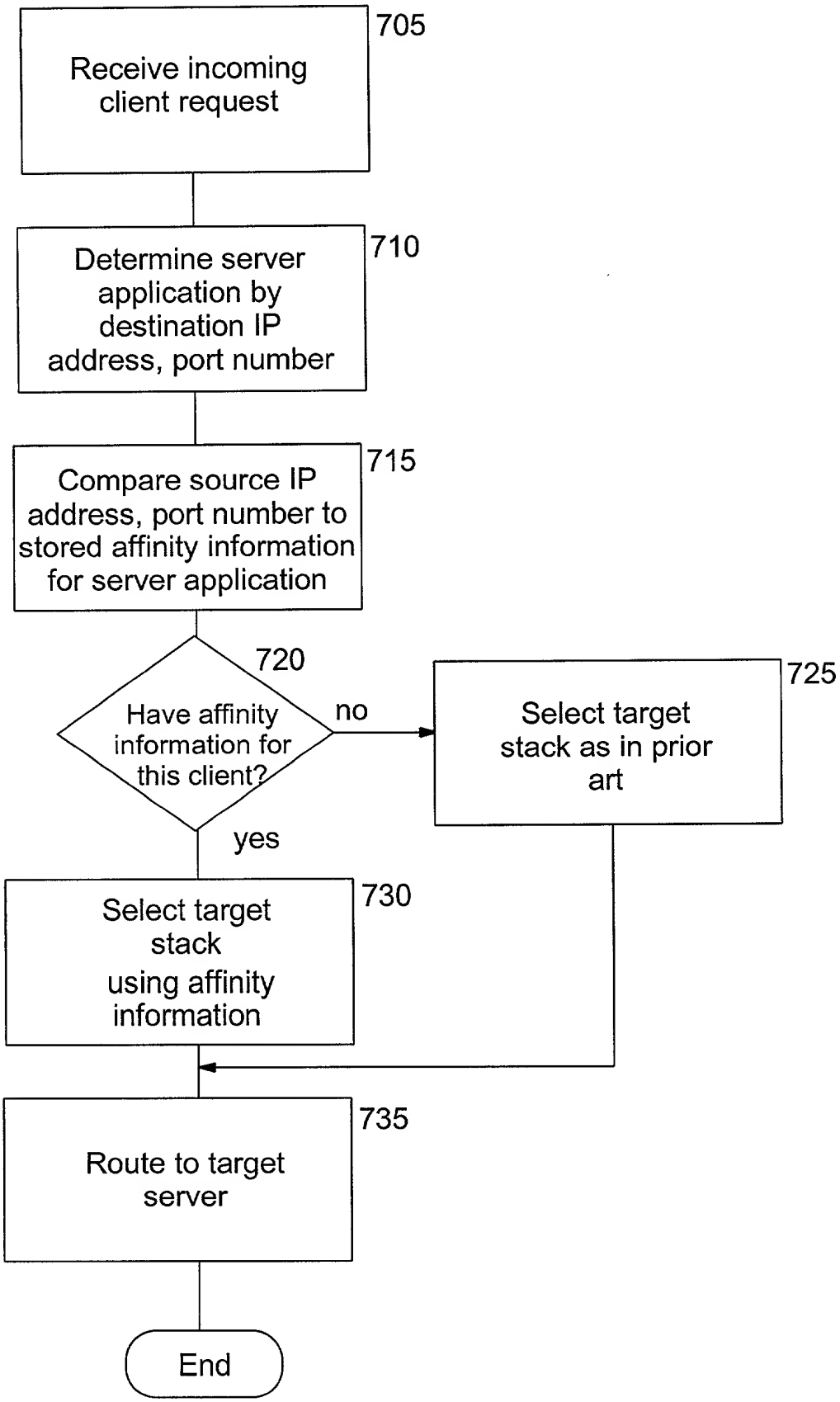

Receive incoming client request
705
Determine server application by destination IP address, port number
710
Compare source IP address, port number to stored affinity information for server application
715
Have affinity information for this client?
720
no
Select target stack as in prior art
725
yes
Select target stack using affinity information
730
Route to target server
735
End

FIG. 8

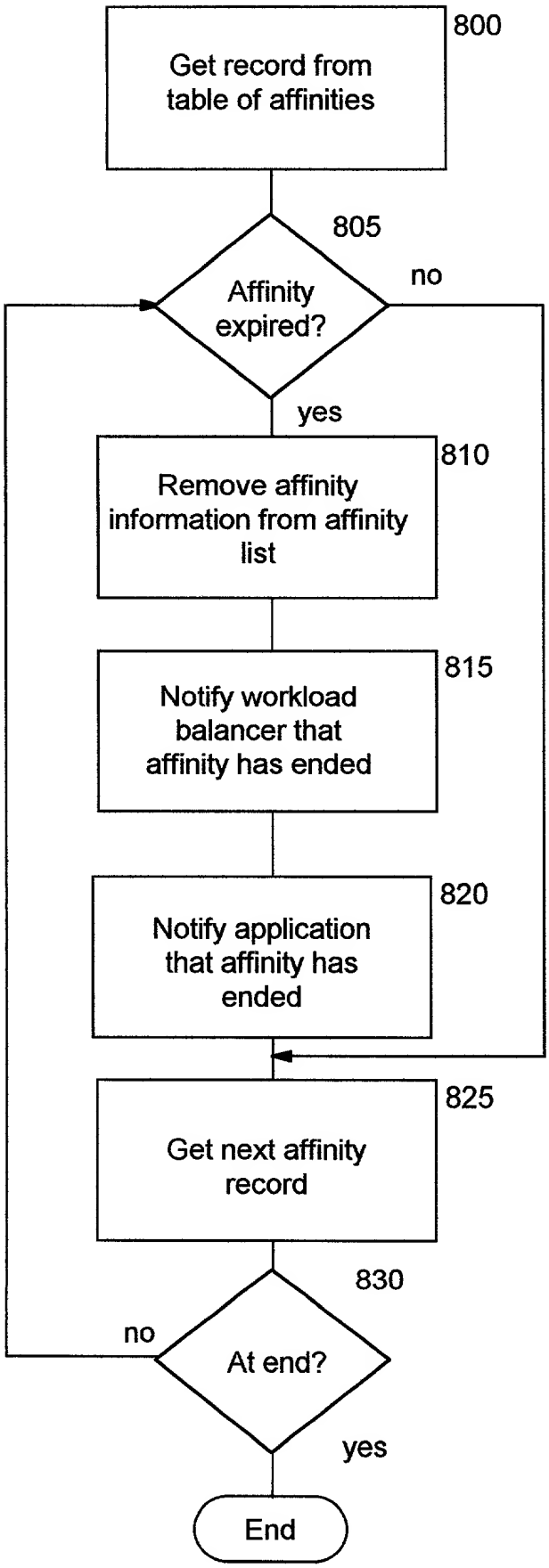

Get record from table of affinities
800
805
Affinity expired?
no
yes
Remove affinity information from affinity list
810
Notify workload balancer that affinity has ended
815
Notify application that affinity has ended
820
Get next affinity record
825
830
At end?
no
yes
End

FIG. 9

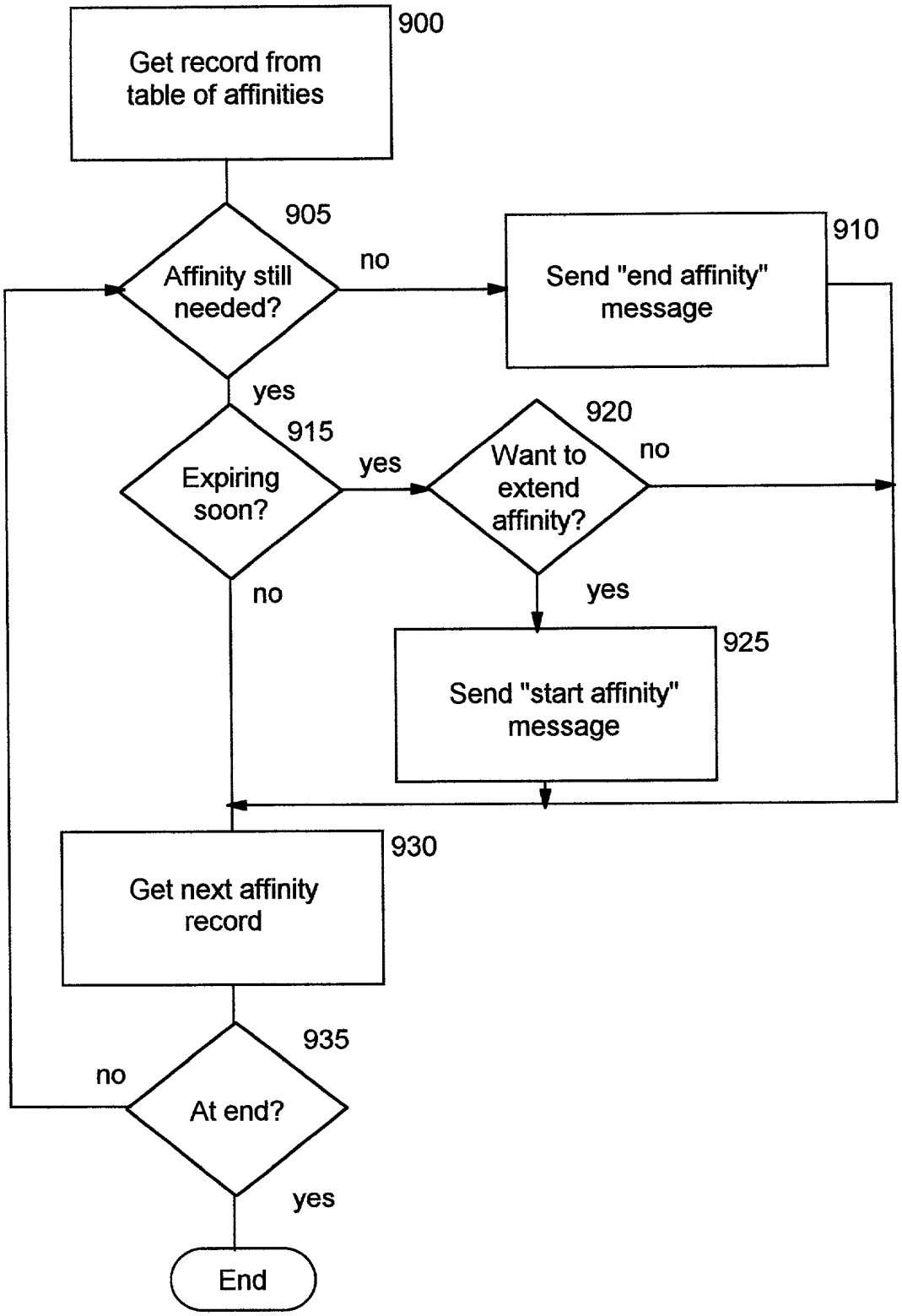

Get record from table of affinities
900
Affinity still needed?
905
no
Send "end affinity" message
910
yes
Expiring soon?
915
yes
Want to extend affinity?
920
no
yes
no
Send "start affinity" message
925
Get next affinity record
930
At end?
935
no
yes
End

FIG. 10

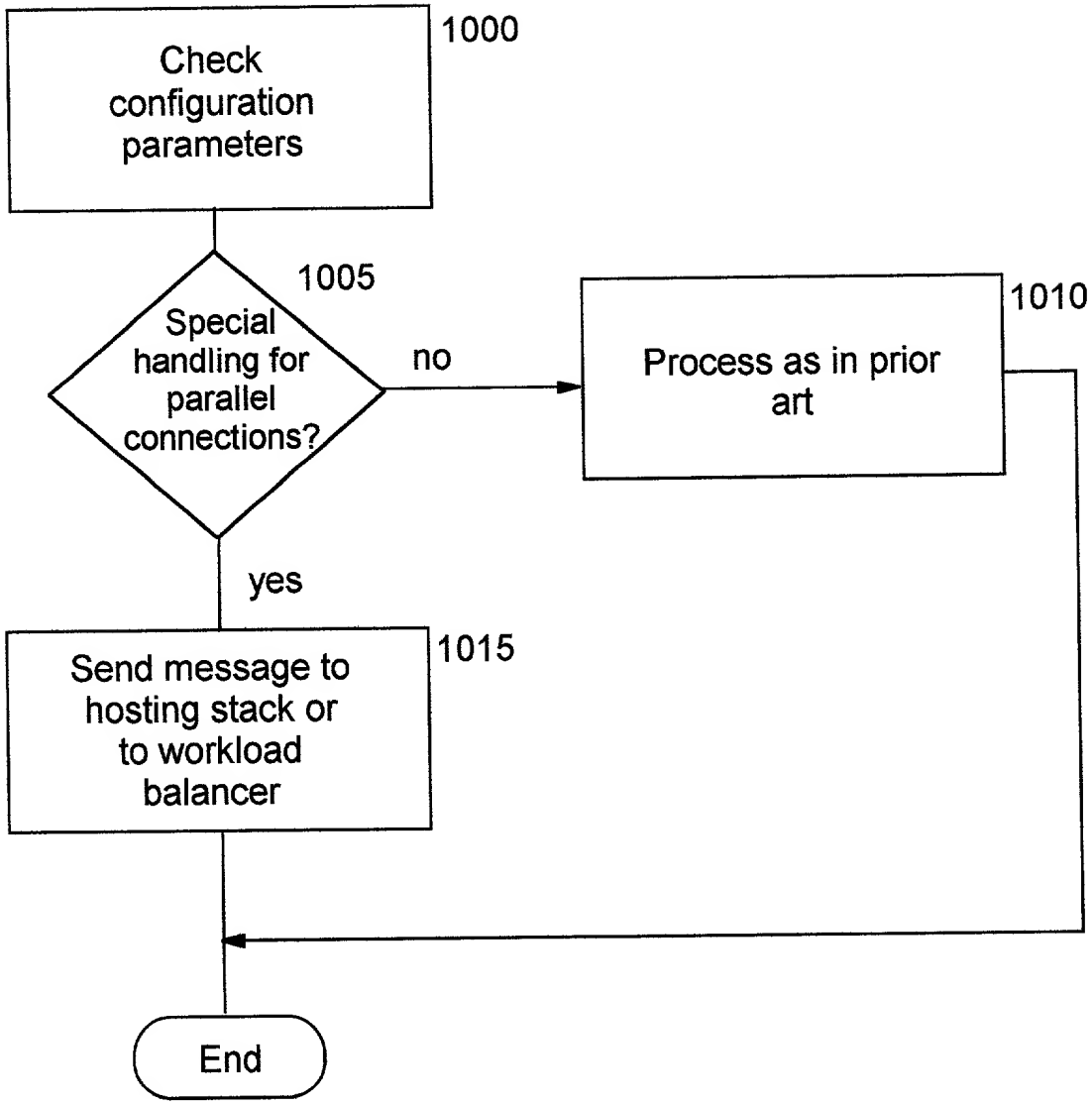
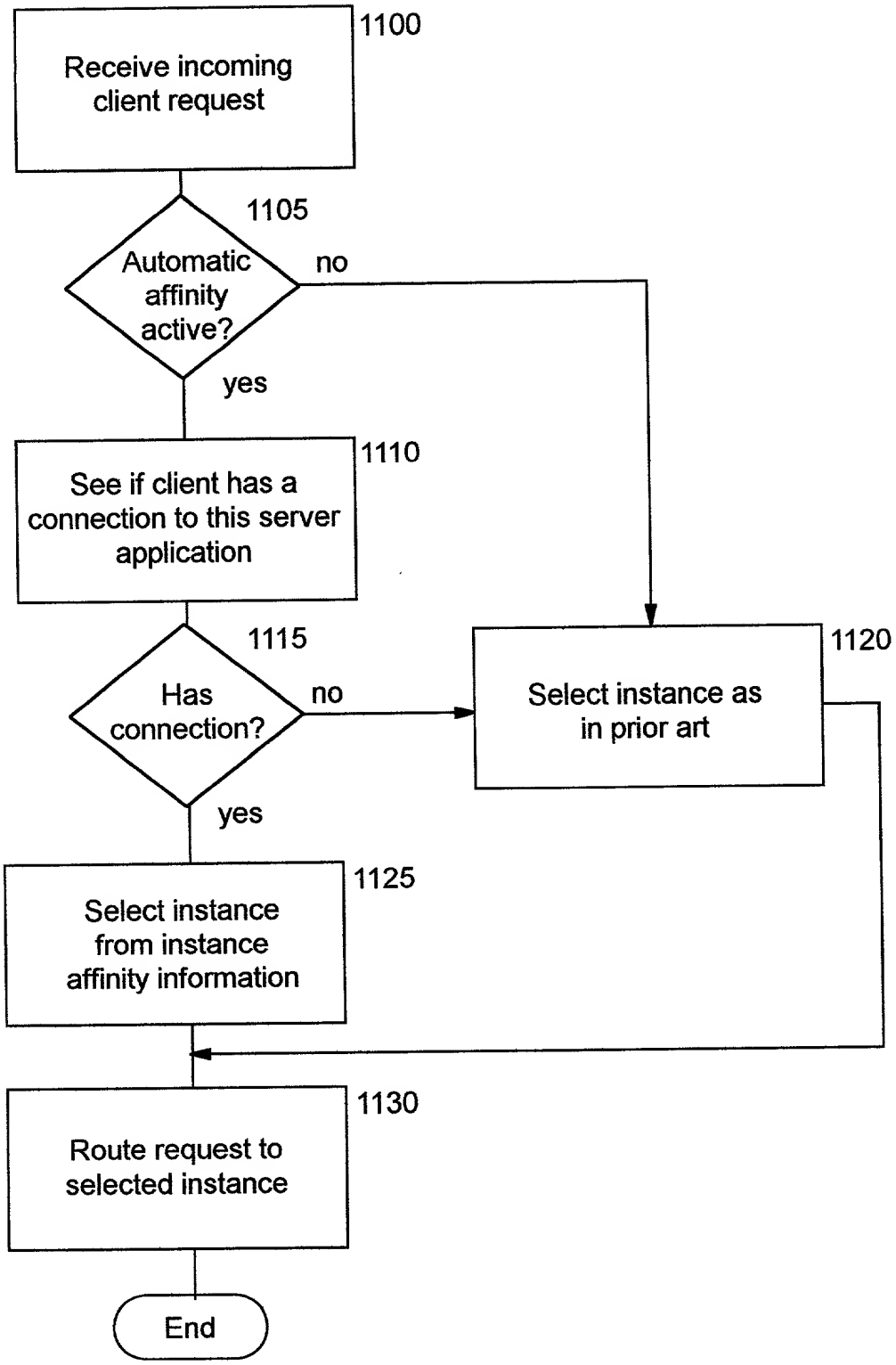

Check configuration parameters
1000
Special handling for parallel connections?
1005
no
Process as in prior art
1010
yes
Send message to hosting stack or to workload balancer
1015
End

FIG. 11


Receive incoming client request
1100
Automatic affinity active?
1105
no
yes
See if client has a connection to this server application
1110
Has connection?
1115
no
yes
Select instance as in prior art
1120
Select instance from instance affinity information
1125
Route request to selected instance
1130
End